This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents *will not* correct images, please do not report the images to the Image Problem Mailbox.

INDEX

Internet Engineering Task Force Audio/Video Transport Working Group
INTERNET DRAFT
Balabanian-Nortel
draft-ietf-avt-rtp-mpeg4-dmif-00.txt
March 11,1998
Expires: September 16,1998

The Role of DMIF in Support of RTP MPEG-4 Payloads

STATUS OF THIS MEMO

This document is an Internet-Draft. Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or made obsolete by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as ``work in progress''.

To learn the current status of any Internet-Draft, please check the ``1id-abstracts.txt'' listing contained in the Internet-Drafts Shadow Directories on ftp.is.co.za (Africa), nic.nordu.net (Europe), munnari.oz.au (Pacific Rim), ds.internic.net (US East Coast), or ftp.isi.edu (US West Coast).

Distribution of this document is unlimited.

ABSTRACT

This memorandum describes how RTP carrying MPEG-4 payloads interacts with the MPEG-4 Access Unit layer through the MPEG (Delivery Multimedia Integration Framework) DMIF. DMIF is used to pass information essential for the packing and unpacking of MPEG-4 streams into RTP as well as for adjusting the MPEG-4 AL-PDU lengths to be within the path MTU.

DMIF interprets the RTCP reports by comparing its statistics to the requested MPEG-4 media based QoS. If the statistics fail to meet the requested QoS then action is taken to either continue with the impaired perforamce, upgrade the network service class, scale down the stream or delete the stream.

This specification is a product of the Audio/Video Transport working group within the Internet Engineering Task Force. Comments are solicited and should be addressed to the working group's mailing list

at rem-conf@es.net and/or the authors.

1 Introduction

MPEG-4 is a recent standard from ISO/IEC for the coding of natural and synthetic audio-visual data in the form of audiovisual objects that are arranged into an audiovisual scene by means of a scene description [1] [2][3][4]. This memorandum specifies how DMIF is used to facilitate the operation of the RTP MPEG-4 payloads [5][6].

This memorandum provides a solution for discussion in IETF AVT and ISO/IEC MPEG technical communities in order to identify issues in using of MPEG-4/DMIF with RTP and incorporate the results. This would lead to the finalization of the specification on RTP use of MPEG-4 with DMIF.

1.1 Overview of MPEG-4 End-System Architecture

Figure 1 below shows the general architecture of MPEG-4 terminals. The Compression Layer processes individual audio-visual media streams without regard to delivery technologies. The compression schemes in MPEG-4 achieve efficient encoding over a wide range from Kbps to multiple Mbps. The MPEG-4 compression schemes are defined in the ISO/IEC specifications 14496-2 and -3 [2][3]. The media content at this layer is organized in Elementary Streams.

The MPEG-4 Systems specification, ISO/IEC 14496-1 [1], defines the concepts needed to describe the relations between Elementary Streams in a way that allows to create distributed, yet integrated, content presentations and to synchronize the streams. This part of the specification is both media unaware and delivery technology unaware.

The Delivery Layer in MPEG-4 consists of the Delivery Multimedia Integration Framework defined in ISO/IEC 14496-6 [4]. This layer is media unaware but delivery technology aware. It provides transparent access to and delivery of content irrespective of the technologies used. The interface between the AU Layer and DMIF is called DMIF Application Interface (DAI). It offers content location independent procedures for establishing MPEG-4 sessions and access to transport channels.

```
media aware          +-----------------------------------------+
delivery unaware     |          COMPRESSION LAYER              |
14496-2 Visual       |streams from as low as Kbps to multi-Mbps|
14496-3 Audio        +-----------------------------------------+
                                                         Elementary
                                                         Stream
==============================================================Interface
                                                           (ESI)
                    +------------------------------------------+
media and           |              SYSTEMS LAYER               |
delivery unaware    | manages elementary streams, their synch- |
14496-1 Systems     | ronization and hierarchical relations    |
                    +------------------------------------------+
                                                          DMIF
                                                        Application
===============================================================Interface
                                                           (DAI)
                    +------------------------------------------+
delivery aware      |              DELIVERY LAYER              |
media  unaware      |provides transparent access to and delivery|
14496-6 DMIF        | of content irrespective of delivery      |
                    |                technology                |
                    +------------------------------------------+
        Figure 1: General MPEG-4 terminal architecture
```

1.2 The DMIF Model

DMIF as an integration framework uses a uniform procedure at the DAI interface to access the MPEG-4 content irrespective whether the content is broadcast, stored on a local file or obtained through interaction with a remote end-system. The model is shown in Figure 2 below.

The specific instance of interest in this memorandum is the interaction with a remote end-system. For this case DMIF uses internal (informative) DMIF-Network Interface(DNI)to map the controls obtained from the application through DAI into the various signaling appropriate

to the various networks.


```
        ! +---+ +-----------+  +-----------+  +-----------+
        ! |   | |Local DMIF |  |Remote DMIF|  |Remote App.|
+-----+ ! | D | |for Brd'cst|  |(locally   |  |(locally   |     Brd'cst
|     | ! | M | |           |  | emulated) |  | emulated) |<-----source
|Local| ! | I | +-----------+  +-----------+  +-----------+
|     | ! | F | +-----------+  +-----------+  +-----------+
|App  | ! |   | |Local DMIF |  |Remote DMIF|  |Remote App.|      File
|     | ! | F | |for Local  |  |(locally   |  |(locally   |<-----source
|     | ! | i | |  Files    |  | emulated) |  | emulated) |
|     | ! | l | +-----------+  +-----------+  +-----------+
|     | ! | t | +-----------+ ! +---+  /     ----      \
|     | ! | e | |Local DMIF | ! |Sig| |   ---      ----  |
+-----+ ! | r | |for Remote | ! |map|<->(   Network   ) |Outside the
        ! |   | |  Service  | ! |   | |   ----   -----  |scope of DMIF
        ! +---+ +-----------+ ! +---+ |     -----       |
       DAI                   DNI      |      ^          |
                                       \_____|_________/
                                             |
                                             |
                                             | +---+!+------+!+------+
                                             | |Sig|!|Remote|!|Remote|
                                             +>|map|!| DMIF |!| App. |
                                               |   |!|(real)|!|(real)|
                                               +---+!+------+!+------+
                                                   DNI      DAI
```

1.3 Mapping between MPEG-4/DMIF and RTP

Figure 3 below draws the correspondence between RTP and MPEG-4/DMIF. It is noted that DAI signaling allows the establishment of an MPEG-4 Service e.g., Video on Demand, the request of channels to carry MPEG-4 Elementary Streams for that service and the reading of Elementary Stream data when received.

```
               RTP                                MPEG-4/DMIF
   +------------------------------+   +----------------------------+
  / DATA TRANSPORT      CONTROL    \ /     DATA TRANSPORT           \
       (RTP)            (RTCP)     |  (Elementary Stream)
    Audio, Video     SR, RR, SDES  .    Audio, Video
    Simulated Data   BYE, APP      |    Simulated Data
                                   .
          ^              ^         |          ^
          I              |         .          I    CONTROL
          I              |         |          I   (Scene, Object
          I              v         .          I    Descriptor(OD))
          I         Corresponds to |          I       ^
          I         DAI SIGNALING  .          I       |
          I                        |          I       |
          I                        .          I      /
          I                        |          I     /
          I                        .          I    /
                                   |          I   /
```

```
   In addition to                   .          I /     SIGNALING
   Audio, Video                     |          I/   (Service, Channel,
   Simulated Data                   .          I          Data)
   to include Scene                 |          I           ^
   and OD streams                   .      ====I==========|=== DAI
                                    |          I          |
                                    .          I          v
                                    |   Elementary Streams
                                    .   in AL-PDU fragments
```

Figure 3: Drawing some correspondence between MPEG-4/DMIF and RTP

The AL-PDUs can be mapped to RTP-PDUs as follows [6]:

```
           RTP-PDU 1:1 AL-PDU
           RTP-PDU 1:N AL-PDU
           RTP-PDU N:1 AL-PDU
```

The selection from above choices is based on the size of the AL-PDU with respect to the RTP-PDU MTU (IP) size. The first choice can use MPEG-4 single stream RTP payload type. The second case can use MPEG-4 FlexMux RTP payload type. The last choice will also use MPEG-4 FlexMux RTP payload type. The last situation occurs when MPEG-4 Access Unit is not able to adjust the MPEG-4 AL-PDU lengths to be within the path MTU.

1.4 MPEG-4 Media Based RTP QoS

The following Media based QoS can be derived for the different mappings.

1.4.1 The case RTP-PDU 1:1 or N:1 AL-PDU

RTP Media QoS_QualifierTag	Derivation from the ES Media transport-QoS
MAX_DELAY of RTP PDU	Maximum delay per AU (microseconds) measured over 1 sec.
AVG_DELAY of RTP PDU	Average delay per AU allowed (microseconds) measured over 1 min
Dejitter Buffer for the RTP stream	Adjusted for the overhead of the RTP PDU
LOSS_PROB of RTP PDU	Probability of loss of any single AU

	(Fraction (0.00 - 1.00) over 1 sec.
MAX_GAP_LOSS of RTP PDUs	Maximum allowable consecutive AU loss measured over 1 sec
MAX_RTP_SIZE	Maximum size of an AU (Bytes) Plus AL-PDU and RTP overhead
MAX_RTP_RATE	Maximum arrival rate of AUs (RTP-PDU/second)
AVG_RTP_SIZE	Average size of AUs (Bytes) Plus AL-PDU and RTP overhead

1.4.2 The case RTP-PDU 1:N AL-PDU

RTP Media QoS_QualifierTag	Derivation from the ES Media QoS
MAX_DELAY of RTP PDU	Least Maximum delay per AU from among the N AL-PDUs (microseconds) measured over 1 sec.
AVG_DELAY of RTP PDU	Average delay per AU allowed (microseconds) measured over 1 min.
Dejitter Buffer for the RTP stream	Total of deitter buffers adjusted for the overhead of the RTP PDU

LOSS_PROB of RTP PDU	Least Probability of loss of any single AU from the N AL-PDUs (Fraction (0.00 - 1.00) over 1 sec.
MAX_GAP_LOSS of RTP PDUs	Least Maximum allowable consecutive AU loss for any one of the N AL-PDUs measured over 1 sec
MAX_RTP_SIZE	Sum of the MAX_AU_SIZEs of from each of the N AL-PDUs Plus AL-PDU and RTP overhead
MAX_RTP_RATE	Highest MAX_AU_RATE of AUs from each of the N AL-PDUs (RTP-PDU/second)
AVG_RTP_SIZE	Sum of Average size of AUs from each of the N AL-PDUs Plus AL-PDU and RTP overhead (Bytes)

Note all the Streams chosen for encapsulation with RTP belong to the same priority level.

2 Operation of the RTP MPEG-4 payloads with DMIF

Figure 4 and 5 show the conceptual operation of the MPEG-4/DMIF with RTP.
The DAI signaling shall be used to set up the MPEG-4 session. When the RTP is used A local (or remote) DMIF entity could be used to start the RTP session with its corresponding RTP data transport (carrying one or more MPEG-4 Elementary Streams) and RTCP for control.

Figure 4 shows that in case of a single MPEG-4 stream payload type, the ALConfigDescriptor is being received at the sender side and being used both at the sender and the receiver for efficient packing of the stream into the RTP transport [6].


```
                     ^
                     |
                     |
                   +---+                        SIGNALING
                   |AL |                 (MTU + ALConfigDescriptor)
                   +---+
                     ^                                 ^
                     |                                 |
==================|=================================|===DAI===
                     |                                 v
+----------------------------------------+      +---------------+
|        Regenerate AL-PDUs              |<------|     DMIF      |
```

```
+-------------------------------------+  .    |                |
 MPEG-4  ^   Time   ^ Sequence^          .    +----------------+
Payloads |  Stamps  |  Numbers|       ALConfig  ^        ^
         |          |         |      Descriptor |        |
       +------------------------+               v        |
       |       RTP Parser       |         +-----------+ |
       +------------------------+         |RTCP Parser| |
                    ^                     +-----------+ |
                    I                           ^       |
                    I                           |       |
                    I                           v       v
                                                ~~~~~~~~DNI
```

Figure 4: Conceptual view of the operation of DMIF with MPEG-4 single stream RTP Payload type

Figure 5 shows that in the case of MPEG-4 FlexMux RTP payload type, information for the MTU and ALConfig is not required. The FlexMux decoder however needs the MuxCode information which is generated at the sending end by the FlexMux muxing code and passed to the receiver through the DMIF signaling.


```
 Audio, Video, Simulated Data, Scene, ODs

  ^     ^     ^                      ^
  |     |     |                      |
  |     |     |                      |
+---+ +---+ +---+                  +---+
```

```
|AL |  |AL |  |AL |.  .  .  .  .  .  |AL |
+---+  +---+  +---+                  +---+                    SIGNALING
  ^      ^      ^                      ^                          ^
  |      |      |                      |                          |
==|====|======|====================|====================|===DAI===
  |      |      |                      |                          v
+--------------------------------------+       +---------------+
|              FlexDemux               |<------|     DMIF      |
+--------------------------------------+   .   |               |
 MPEG-4   ^      Time    ^ Sequence^            .   +---------------+
Payloads |     Stamps  |  Numbers|           .       ^       ^
         |             |         |   MuxCode        |       |
        +------------------------+    Descriptor   v       |
        |       RTP Parser       |             +------------+|
        +------------------------+             |RTCP Parser||
                      ^                        +------------+|
                      I                             ^       |
                      I                             |       |
                      I                             v       v
                                                    ~~~~~~~~DNI
```

Figure5: Conceptual view of the operation of DMIF with MPEG-4 FlexMux RTP Payload type

3 RTCP Sender and Receiver Reports

RTP receivers provide reception quality feedback using RTCP report packets which may take one of two forms depending upon whether or not the receiver is also a sender.

These reports shall be used by DMIF in the case of MPEG-4/DMIF-RTP to readjust the demand put on the network based on a predefined policy which may involve a decision to be made by the user.

The sender report packet consists of three sections, possibly followed by a fourth profile-specific extension section if defined (none has been specified so far for MPEG-4 RTP payloads).

The third section contains zero or more reception report blocks depending on the number of other sources heard by this sender since the last report. Each reception report block conveys statistics on the reception of RTP packets from a single synchronization source.

SSRC_n (source identifier):
The SSRC identifier of the source to which the information in this reception report block pertains. This SSRC may either relate to an MPEG-4 single or FelxMux RTP payload.

fraction lost:

The fraction of RTP data packets from source SSRC_n lost since the previous SR or RR packet was sent, expressed as a fixed point number.

This fraction is compared to the LOSS_PROB. It is important that the duration over which this metric is measured corresponds to the same duration used to express the LOSS_PROB. If the statistics consistently exceeds the LOSS_PROB then the policy enforcer is brought into action. As a result the load on the RTP stream is reduced.

Note: RTCP does not provide a measure for consecutive RTP-PDU loss in order to be able to calculate the GAP_LOSS.

interarrival jitter:
An estimate of the statistical variance of the RTP data packet interarrival time, measured in timestamp units and expressed as an unsigned integer.

The jitter calculation in RTCP is based on the variation of consecutive interarrival times:

If Si is the RTP timestamp from packet i, and Ri is the time of arrival in RTP timestamp units for packet i, then for two packets i and j, D may be expressed as

$$D(i,j)=(Rj-Ri)-(Sj-Si)=(Rj-Sj)-(Ri-Si)$$

The interarrival jitter is calculated continuously as each data packet i is received from source SSRC_n, using this difference D for that packet and the previous packet i-1 in order of arrival (not necessarily in sequence), according to the formula

$$J=J+(|D(i-1,i)|-J)/16$$

Whenever a reception report is issued, the current value of J is sampled.

The J value must be converted to msecs J'

J' must be <= to 500*Dejitter_Buffer/(MAX_RTP_SIZE* MAX_RTP_RATE)

If this value is exceeded consistently for some time then the QoS policy enforcer is brought into action. As a result the load on the RTP stream is reduced.

Round trip Delay:
This delay is calculated by measuring the time sending a sender report and receiving the associated receiver report and subtracting the delay it took to send the receiver report at the receiver.

Delay must be <= .5*MAX_DELAY converted to seconds from microseconds

Average Delay over 1 minute <= .5*AVG_DELAY converted to seconds from microseconds

If either of the above values is exceeded consistently for some time then the QoS policy enforcer is brought into action. As a result the load on the RTP stream is reduced.

4. BYE: Goodbye RTCP packet
When a single stream in case of the MPEG-4 single stream RTP payload, a BYE packet is sent along with the DS_ChannelDelete using DMIF-DMIF signaling. At the receiver either a BYE packet or DS_ChannelDelete signal will cause DAI to pass DA_ChannelDelete to the application. When a FlexMux stream is used, the BYE packet is generated when no longer any MPEG-4 streams are carried on the RTP session. This means that DS_ChannelDeletes have already been sent for all the channels carried on the RTP session and the application has been notified by DA_ChannelDelete(s) across the DAI. A BYE message is followed by a DS_TransMuxDelete which at the reception will allow both the sending and receiving DMIF sides to reuse the RTP/IP ports.

A Authors' Addresses

Vahe Balabanian
Nortel
P.O.Box 3511, St. C
Ottawa, Ontario
Canada K1Y 4H7
Email: balabani@nortel.ca

B Bibliography

[1] ISO/IEC 14496-1 CD 'MPEG-4 Systems' Oct. 1997

[2] ISO/IEC 14496-2 CD 'MPEG-4 Visual' Oct. 1997

[3] ISO/IEC 14496-3 CD 'MPEG-4 Audio' Oct. 1997

[4] ISO/IEC 14496-6 CD 'Delivery Multimedia Integration Framework',Oct. 1997

[5] Schulzrinne, Casner, Frederick, Jacobson 'RTP: A Transport Protocol for Real Time Applications' RFC 1889,Internet Engineering Task Force, Jan. 1996.

[6] Carsten, Balabanian, Basso, Civanlar, hoffman, Speer, Schulzrinne, 'RTP payload format for MPEG-4 Elementary Streams' draft-ietf-avt-rtp-mpeg4, Internet Engineering Task Force, March 1998.

March 11,1998
Expires: September 16,1998

Original

INDEX

Internet Engineering Task Force Audio/Video Transport Working Group
INTERNET DRAFT
Balabanian-Nortel
draft-balabanian-rtsp-mpeg4-dmif-00.txt
Sept. 22,1998
Expires: March 26,1999

The Role of DMIF with RTSP and MPEG-4

STATUS OF THIS MEMO

This document is an Internet-Draft. Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or made obsolete by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as ``work in progress''.

To learn the current status of any Internet-Draft, please check the ``1id-abstracts.txt'' listing contained in the Internet-Drafts Shadow Directories on ftp.is.co.za (Africa), nic.nordu.net (Europe), munnari.oz.au (Pacific Rim), ftp.ietf.org (US East Coast), or ftp.isi.edu (US West Coast).

Distribution of this document is unlimited.

ABSTRACT

This draft technical proposal describes how MPEG DMIF (Delivery Multimedia Integration Framework) can be used with RTSP for setting MPEG-4 service sessions and subsequently detaching the service. DMIF creates an instance of DMIF RTSP augmented with QoS appropriate for each MPEG-4 stream and also makes use of the FlexMux to carry multiple streams on a single UDP or TCP IP flow. The stream control play/pause etc. using RTSP is executed directly by the MPEG-4 Systems Sync Layer.

Comments are solicited and should be addressed to the working group's mailing list at confctrl@isi.edu and/or the authors.

1 Introduction

MPEG-4 is a recent standard from ISO/IEC for the coding of natural and synthetic audio-visual data in the form of audiovisual objects that are arranged into an audiovisual scene by means of a scene description [1] [2][3][4]. This draft proposal specifies how DMIF is used with RTSP [5

and RTP MPEG-4 payloads over Internet[6][7][8].

This draft proposal provides a solution for discussion in IETF MMUSIC and ISO/IEC MPEG technical communities in order to identify issues in using MPEG-4/DMIF with RTSP and will incorporate the results in the MPEG-4specifications and issue this proposal as an RFC draft.

The MPEG-4 standards are versioned. Each version beyond V1 represents a backward compatible extension. MPEG-4 V1 is targeted to become ISO International Standard on December 1998 and each subsequent version wil be displaced approximately by a year. MPEG-4 V2 is targeted for Februar 2000. DMIF is part 6 of the MPEG-4 standard.

This technical proposal will impact on MPEG V2 International Standard targeted for February 2000.

The content of this draft technical proposal is intentionally kept at a tutorial level in order to facilitate the discussion among the interested participants.

1.1 Overview of MPEG-4 End-System Architecture

Figure 1 below shows the general architecture of MPEG-4 terminals. The Compression Layer processes individual audio-visual media streams without regard to delivery technologies. The compression schemes in MPEG-4 achieve efficient encoding over a wide range from few Kbps to multiple Mbps. The MPEG-4 compression schemes are defined in the ISO/IEC specifications 14496-2 and -3 [2][3]. The media content at this layer is organized in Elementary Streams.

The MPEG-4 Systems specification, ISO/IEC 14496-1 [1], defines the concepts needed to describe the relations between Elementary Streams in a way that allows the creation of distributed, yet integrated, content presentations and to synchronize the streams. This part of the specification is both media unaware and delivery technology unaware.

The Delivery Layer in MPEG-4 consists of the Delivery Multimedia Integration Framework defined in ISO/IEC 14496-6 [4]. This layer is media unaware but delivery technology aware. It provides transparent access to and delivery of content irrespective of the technologies used. The interface between the Sync Layer and DMIF is called DMIF Application Interface (DAI). It offers content location independent procedures for establishing MPEG-4 sessions and access to transport channels. DMIF is primarily an integration framework. It provides a default DMIF signaling (DS) protocol which corresponding to DMIF Networ Interface (DNI), see Figure 2. DS is used to complement the lack functionality in underlying control protocols in order to keep the integrity of the DMIF framework.

```
media aware          +-----------------------------------------+
delivery unaware     |             COMPRESSION LAYER           |
14496-2 Visual       |streams from as low as Kbps to multi-Mbps|
14496-3 Audio        +-----------------------------------------+
                                                               Elementary
                                                               Stream
==============================================================Interface
                                                                 (ESI)
                    +-------------------------------------------+
media and           |             SYSTEMS SYNC LAYER            |
delivery unaware    | manages elementary streams, their synch-  |
14496-1 Systems     | ronization and hierarchical relations     |
                    +-------------------------------------------+
                                                                DMIF
                                                              Application
===============================================================Interface
                                                                 (DAI)
                    +-------------------------------------------+
delivery aware      |                DELIVERY LAYER             |
media  unaware      |provides transparent access to and delivery|
14496-6 DMIF        | of content irrespective of delivery       |
                    |               technology                  |
                    +-------------------------------------------+
```

Figure 1: General MPEG-4 terminal architecture

1.2 The DMIF Model

DMIF as an integration framework uses a uniform procedure at the DAI interface to access the MPEG-4 content irrespective whether the content is broadcast, stored on a local file or obtained through interaction with a remote end-system. The model is shown in Figure 2 below.

The specific instance of interest in this memorandum is the interaction with a remote end-system. For this case DMIF uses internal (informative) DMIF-Network Interface(DNI)to map the controls obtained from the application through DAI into the various signaling appropriate to the various networks. The default end-to-end DMIF signaling (DS) protocol which corresponds to DNI is specified in DMIF V1 [4].

```
        ! +---+ +-----------+  +-----------+  +-----------+
        ! |   | |Local DMIF |  |Remote DMIF|  |Remote App.|
+------+! | D | |for Brd'cst|  |(locally   |  |(locally   |     Brd'cst
|      |! | M | |           |  | emulated) |  | emulated) |<-----source
|Local |! | I | +-----------+  +-----------+  +-----------+
|      |! | F | +-----------+  +-----------+  +-----------+
|App   |! |   | |Local DMIF |  |Remote DMIF|  |Remote App.|      File
|      |! | F | |for Local  |  |(locally   |  |(locally   |<-----source
|      |! | i | |  Files    |  | emulated) |  | emulated) |
|      |! | l | +-----------+  +-----------+  +-----------+
|      |! | t | +-----------+ ! +---+  /     ----      \
|      |! | e | |Local DMIF | ! |Sig| |   ---     ----   |
+------+! | r | |for Remote | ! |map|<->(   Network    ) |Outside the
        ! |   | |  Service  | ! |   | |   ----   -----   |scope of DMIF
        ! +---+ +-----------+ ! +---+ |      -----       |
       DAI                   DNI      |        ^         |
                                      \________|_________/
                                               |
                                               |
                                                 +---+!+-------+!+-------+
                                                 |Sig|!|Remote |!|Remote |
                                               +>|map|!| DMIF  |!| App.  |
                                                 |   |!|(real) |!|(real) |
                                                 +---+!+-------+!+-------+
```

```
                                                DNI      DAI
```

Figure 2: The DMIF model covers broadcast, local file storage and remote service with a uniform procedure for application transparency

2 Placing RTSP within the MPEG-4 architecture

In order to make use of the functions enabled by RTSP, DMIF RTSP Client and Server modules are created as shown in Figure 3 below. These modules interface with DAI and interwork with the MPEG-4 Sync Layer. It is an internal implementation decision not affecting interoperability whether to combine the DRTSP client and server functions with the Sync Layer or keep them separate. It is noted that DAI allows the establishment of an MPEG-4 service session e.g., RTSP Video on Demand session, the request of channels to carry MPEG-4 Elementary Streams for that service are executed by the DMIF-RTSP (DRTSP) module instantiated by DMIF as a response to the DRTSP-URL in the MPEG-4 service session request. This module as described here only carries out RTSP SETUP and TEARDOWN functions while the stream control functions are carried out through the interactions of the DRTSP servers with the Sync layer.

The signaling between the two instances of the DRTSP at the originating and destination DMIFs can initially use the default DMIF signaling and wrap the RTSP related fields in a DMIF-to-DMIF Data fields. Later versions may extend the RTSP to include the DMIF functionality, in that case the proper RTSP signaling will be used. Backward compatibility is assured by the way of a common set of DMIF Network Interface (DNI) primitives, see Figure 2 which are used to generate the DMIF signaling messages or their RTSP extensions.

```
               DAI                            DAI
 +-----+       I                               I       +-----+
/ DRTSP \      I+-----+ DMIF Signaling +-----+I       / DRTSP \
```

```
| Client   |<--->|Orig.|<-------------->|Dest.|<--->| Server   |
 \       /      I|DRTSP|                |DRTSP|I     \        /
  +-----+       I+-----+                +-----+I      +-----+
     ^          I                              I         ^
     |          I                              I         |
     |          I                              I         |
     v          I                              I         v
  +-----------+I                               I +-----------+
  | MPEG-4    |I                               I | MPEG-4    |
  |Systems    |<===============================> |Systems    |
  |Sync Layer |I                               I |Sync Layer |
  +-----------+I                               I +-----------+
```

Figure 3: RTSP within the MPEG-4 architecture

3 DMIF RTSP Message Sequences

3.1 Setting up MPEG-4 RTSP service session

MPEG-4 is an object based encoding with Scenes described in Binary Information Formatted Streams (BIFS)and objects placed within a scene described with Object Descriptor (OD) streams [1]. In order to be able to begin viewing MPEG-4, both the BIFS and the OD decoders must be set for parsing at the receiver end in order to be able to decode the BIFS and OD streams[1]. This information is obtained from the Initial Object Descriptor (IOD)file. The location of IOD can be expressed in a DRTSP-URL which can be obtained by any means. The DRTSP URL indicates that the MPEG-4 play control functions use the RTSP controls as defined in RFC 2326 [5].

As a result of this action DMIF instantiates DRTSP instances at both the originating and destination DMIF locations and engages the DRTSP client with the DRTSP server. The signaling between the two DRTSP instances can be initially carried in DMIF Signaling with RTSP information wrapped in the DMIF-to-DMIF data field. At a later date when RTSP extensions for DMIF are adopted,

```
    DRTSP        DMIF                            DMIF         DRTSP
    Client       DRTSP                           DRTSP        Server
```

```
             DAI  (Orig) DNI                 DNI (Dest) DAI
              I           I                   I          I
              I           I                   I          I
DA_ServiceAttach          I DN_SessionSetup   I          I
       -1-------->0-2---------------------------->       I
(IN:RTSP-URL, I           I  (IN:nsId,        I          I
 uuData)      I           I   calledAddr,     I          I
              I           I   callingAddr,    I          I
              I           I   capDescr)       I          I
              I           I                   I          I
              I           I   (OUT:rsp,       I          I
              I           I        capDescr)  I          I
              I      <--------------------------3-       I
              I           I                   I          I
              I           I DN_ServiceAttach  I    DA_ServiceAttach
              I       -4------------------------->0-5-------->
              I           I(IN:nsId,serviceId,I  (IN:nsId,serviceName,
              I           I serviceName,      I     uuData)
              I           I ddData)           I          I
(OUT:rsp,     I           I                   I          I
 ssId,uuData(IOD))        I (OUT:rsp,ddData)  I  (OUT:rsp,uuData(IOD))
     <-----------8-0<------------------------7-0<-------6-
              I           I                   I          I
```

Figure 4: Command sequence for the establishment of MPEG-4 Service Session

RTSP signaling could be used instead. Figure 4 avoids this format issue by showing the DMIF-NetworkInterface (DNI) primitives instead. This in turn could be mapped into DMIF signaling or to RTSP DMIF extensions.

Step 1
The application in the originating DMIF passes DA_ServiceAttach() indicating the DRTSP-URL and additional User-to-User data "uuData()" e.g., indicating client credentials. DMIF parses the DRTSP-URL and instantiates the originating DRTSP module. The "D" in DRTSP indicates that RTSP is complemented with DMIF signaling.

Step 2
DRTSP strips the <serviceName> from the DRTSP-URL for later use. For the application it assigns a locally significant unique serviceSessionId "ssId". For the network it assigns a globally unique networkSessionId "nsId". It extracts the capabiliyDescriptor "capDescr" which indicates an MPEG-4 service using the DRTSP protocol and contains the identification of the FlexMux [1][6] and any standard protection stacks.

Step 3

When the destination DRTSP receives the the DN_SessionSetup(), both the originating and destination DRTSP peers have knowledge of the same networkSessionId. The compatibility is verified and the appropriate reply is returned identifying the common set in the preferred order of choice.

Step 4
The originating DRTSP assigns a serviceId which is unique for the particular networkSessionId and passes the DN_ServiceAttach() to the destination DMIF indicating the serviceName it has previously stripped from the DRTSP-URL, and additional data ddData which contains the uuDat provided by the application.

Step 5
The destination DRTSP when it receives the DN_ServiceAttach(), it determines the Executive process managing the services, assigns a locally unique serviceSessionId and then sends a DA_ServiceAttach() to it containing the serviceSessionId along with the serviceName and the uuData from the receiver application. The mechanism used in the destination DRTSP to identify the process running the service and to deliver the message to it is outside the scope of the DMIF specifcation [4].

Step 6
The DRTSP Server interprets the uuData and potentially performs tests on client credentials. Then it replies with uuData (which in the case of MPEG-4 contains the Initial OD) and a response code.

The destination DRTSP maintains a table associating the locally

meaningful <serviceSessionId> and the network wide meaningful tuple <networkSessionId, serviceId>

Step 7
The destination DRTSP replies to the DN_ServiceAttach() through the DNI, providing a response code and ddData that contains the uuData provided by the Remote Application.

Step 8
The originating DRTSP uses the locally significant unique serviceSessionId and then replies to the DA_ServiceAttach() with the serviceSessionId, a response code and the uuData originally set by the DRTSP Server. The Local DMIF Layer maintains a table associating the locally meaningful <serviceSessionId> and the network wide meaningful tuple <networkSessionId, serviceId>

3.2 Establishment of channels

This follows MPEG-4 service session establishment shown in section 3.1 during which the compatibilty of the transport stacks are ensured and the Initial Object Descriptor (IOD) information is obtained. From the IOD information different MPEG-4 streams can be requested with their associated traffic load and QoS descriptor. For example in order to establish a reliable channel for stream control, reliable two way channels are requested and in order to carry a stream, be it BIFS or OD stream or a content stream, the specific traffic load and QoS are obtained through the ES_Descriptors sent in the IOD file or the OD stream [1].

```
      DRTSP        DMIF                          DMIF         DRTSP
      Client       DRTSP                         DRTSP        Server
             DAI  (Orig) DNI                  DNI (Dest) DAI
              I           I                    I          I
              I           I                    I          I
DA_ChannelAdd             I  DN_ChannelAdd     I DA_ChannelAdd
       -1-------->0-2-------------------------->0-3-------->
(IN:ssId,     I           I  (IN:nsId,         I (IN:ssId,
 DRTSP URL,   I           I   serviceId,       I  DRTSP URL,loop(chId,
 loop(qos,dir,I           I   loop(CAT,sAdd,   I  qos,dir,sAdd,rAdd,
 uuData(      I           I   rAdd,qos,        I  uuData(
 RTSPvx.x,    I           I   ddData)          I  RTSPvx.x,
 Cseq)))      I           I                    I  Cseq)))
              I           I                    I          I
(OUT:ssId,    I           I                    I (OUT:ssId,
 loop(chId,rspI           I                    I  loop(rsp,
 uuData(      I           I                    I  uuData(
 RTSPvx.x,    I           I                    I  RTSPvx.x,
 SC,Cseq,     I           I  (OUT: loop(       I  SC,Cseq,
 Date, Ssn))) I           I   rsp,ddData))     I  Date, Ssn)))
      <-----------6-0<-------------------------5-0<--------4-
              I           I                    I          I
```

Figure 5: Command sequence for establishment of channels

This function corresponds to the RTSP SETUP with the difference that it is complemented with traffic load and QoS descriptors required for the streams.

Step 1
The DRTSP Client passes a DA_ChannelAdd() indicating the channels it requires. The primitive contains the DRTSP-URL, the serviceSessionId "ssId" for which the Channels are requested. Each channel is characterized by a QoS parameter, a direction parameter "dir (DOWNSTREAM in this case) for the stream and by uuData. containing the RTSP SETUP parameters

Step 2
The originating DRTSP assigns a channelAssociationTag for each

requested channel. It then forwards the request to the peer by passing the DN_ChannelAdd() through the DNI, containing the network-wide unique tuple <networkSessionId, serviceId> corresponding to the serviceSessionId, and for each requested channel its CAT, see Annex A1, the (optional) senderAddress "sAdd" extracted from the DRTSP-URL, the (optional) receiverAddress "rAdd", the (optional) qosDescriptor and associated ddData (which conveys the original uuData).

Step 3
The destination DRTSP receives the DN_ChannelAdd() from the DNI; assigns a locally unique channelHandle "chId" for each requested channel and then issues a DA_ChannelAdd() to the destination Application, containing the locally unique serviceSessionId corresponding to the tuple <networkSessionId, serviceId>, and for each requested channel its locally unique channelHandle, the QoS parameter, the direction parameter (DOWNSTREAM in this case), senderAddress , receiverAddress and associated uuData (which conveys the original uuData). At this point the receiver DMIF is able to associate the locally unique channelHandle to the end-to-end significant, networkSession unique CAT.

Step 4
The DRTSP Server interprets the uuData to determine what stream is actually being requested and checks the availability of such a stream. It then replies with a response code for each requested channel along with uuData containing the RTSP SETUP return parameters.

Step 5
The destination DRTSP replies to the original DN_ChannelAdd() providing for each channel TAT see Annex A1, and further ddData that characterize it, along with a response code. In particular ddData would contain in this case further information on how a particular channel is flexmultiplexed in the TAT, that is in the case of MPEG-4 FlexMux, it provides the FlexMux Channel Number. At this point the destination DRTS is able to associate the CAT to networkSessionId, serviceId, TAT and further flexmultiplexing configuration. ddData may also contain uuData returned by the sender application.

Step 6
The originating DRTSP receives the reply to the DN_ChannelAdd(), assigns a locally unique channelHandle "chId" for each requested channel, and replies to the original DA_ChannelAdd() by providing for each requested channel its channelHandle, uuData and a response code. A this point the originating DRTSP is able to associate the locally uniqu channelHandle to the end-to-end significant, networkSession unique CAT and to associate the CAT to networkSessionId, serviceId, TAT and furthe flexmultiplexing configuration.

3.3 Controlling of MPEG-4 Streams

The control of streams is carried out over reliable channels establishe to transport the RTSP Play and pause commands extended for MPEG-4. This function is carried out in the MPEG-4 Systems Sync Layer [1] and is outside the scope of this memorandum.

3.4 Deletion of channels

Application may request through DAI the deletion of channels for stream that are no longer required. This function corresponds to RTSP TEARDOWN with the difference that the deletion of all channels through DAI does not by itself result in the termination of the MPEG-4 service session. For this to happen the session detachment is required as shown in section 3.5

```
      DRTSP        DMIF                           DMIF         DRTSP
      Client       DRTSP                          DRTSP        Server
             DAI  (Orig) DNI                  DNI (Dest) DAI
              I           I                    I          I
              I           I                    I          I
DA_ChannelDelete          I  DN_ChannelDelete  I DA_ChannelDelete
       -1-------->0-2------------------------>0-3-------->
(IN: loop(    I           I  (IN:nsId,         I (IN:ssId,
 chId,reason  I           I   loop(CAT,reason, I  chId,reason
 uuData(      I           I   ddData)          I  uuData(
 RTSPvx.x,    I           I                    I  RTSPvx.x,
 Cseq,Ssn)))  I           I                    I  Cseq,Ssn)))
              I           I                    I          I
              I           I                    I          I
              I           I                    I                  I
 (OUT:        I           I                    I (OUT:
 loop(rsp,    I           I                    I  loop(rsp,
 uuData(      I           I                    I  uuData(
 RTSPvx.x,    I           I  (OUT: loop(       I  RTSPvx.x,
 SC,Cseq)))   I           I   rsp,ddData))     I  SC,Cseq)))
      <----------6--0<-----------------------5--0<--------4-
              I           I                    I          I
```

Figure 6: Command sequence for the deletion of channels

Step 1
The DRTSP Client passes a DA_ChannelDelete() indicating the channels it wants to delete. The primitive contains the channelHandle(s) along with reason codes.

Step 2

The originating DRTSP stops the actual delivery of data on the indicated Channel(s). The Local DMIF Layer forwards the request to the peer by passing the DN_ChannelDelete() containing the network-wide unique networkSessionId, and for each requested channel its CAT and the reason code.

Step 3
The destination DRTSP receives the DN_ChannelDelete() and issues a DA_ChannelDelete() to the DRTSP Server, containing for each requested channel its locally unique channelHandle and the reason code.

Step 4
The DRTSP Server stops the actual delivery of data on the indicated Channel(s), and replies with response codes. At this point channelHandle(s) are invalidated at the DRTSP Server.

Step 5
The destination DRTSP replies to the original DN_ChannelDelete() providing for each channel a response code. At this point channelHandle(s) and CAT(s) are invalidated at the destination DRTSP.

Step 6
The destination DRTSP replies to the original DA_ChannelDelete() by providing for each channel a response code. At this point channelHandle(s) and CAT(s) are invalidated at the originating DRTSP. The DRTSP Client receives the reply and invalidates the channelHandle(s).

3.5 Detaching the MPEG-4 Service Session

Following the steps for the release of channels the end-user may decide to detach the MPEG-4 service session. The sequence of steps shown in Figure 7 below are followed in order to release the MPEG-4 service session.


```
      DRTSP        DMIF                          DMIF         DRTSP
      Client       DRTSP                         DRTSP        Server
             DAI   (Orig) DNI                DNI (Dest) DAI
```

```
                I           I                    I          I
                I           I                    I          I
DA_ServiceDetach            I DN_ServiceDetach   I    DA_ServiceDetach
       -1--------->0-2---------------------------->0-3-------->
(IN:ssId,       I           I(IN:nsId,serviceId,I    (IN:ssId,
 reason)        I           I reason)           I     reason)
                I           I                    I          I
                I           I                    I          I
(OUT:rsp)       I           I (OUT:rsp)          I     (OUT:rsp)
        <-----------6-0<-----------------------5--0<-------4-
                I           I                    I          I
```

Figure 7: Command sequence for detaching the MPEG-4
Service Session

Step 1
The DRTSP Client passes a DA_ServiceDetach() indicating
the service it wants to terminate. The primitive
contains the serviceSessionId along with a reason code.

Step 2
The originating DRTSP Layer passes a DN_ServiceDetach() indicating the
service it wants to terminate (which is now identified by the tuple
<networkSessionId "nsId", serviceId> along with a reason code.

Step 3
The destination DRTSP receives the DN_ServiceDetach() and passes a
DA_ServiceDetach() to the DRTSP Server indicating the service that
must be terminated (which is now identified by the locally meaningful
serviceSessionId) along with a reason code.

Step 4
The DRTSP Server stops the provision of the service, and
frees all resources used for it. It then replies to the
DA_ServiceDetach() providing a response code. At this point
the locally meaningful serviceSessionId is no longer valid.

Step 5
The destination DRTSP replies to the DN_ServiceDetach() along with the
response code. At this point the network session unique serviceId is no
longer valid.

Step 6
The originating DRTSP replies to the DA_ServiceDetach() along with
the response code. At this point the locally meaningful
serviceSessionId is no longer valid.

4 Open Issues

This section contains the issues that require resolution in this memorandum.

1- This draft technical proposal has only included SETUP and TEARDOWN functions. Do these represent a sufficient set for initial session implementations notwithstanding commands such as Play/Pause etc.?

2- What impact is foreseen from the separation of the RTSP session/connection control commands from the stream control commands?

3- Have the RTSP SETUP and TEARDOWN been properly represented in this draft technical proposal?

4- Is there an equivalent for establishing MPEG-4 Service Session in RTSP? What is it?

5- Does RTSP plan to add traffic load and QoS descriptors for the streams in a generic fashion expressed through parameters?

A DMIF Definitions and Symbols

A.1 Definition(s)

Association Tag: In the context of this specification an Association Tag is used to identify elements within a network session with unique end-to-end significant values.

Channel: Is the entity over which a DMIF User sends or receives data.

DMIF-Application Interface: Is the interface between an application (DMIF User) and the DMIF Layer.

DMIF Instance: Is a component in the DMIF layer which deals with a specific delivery technology. It ensures interoperability between DMIF terminals situated on this delivery technology.

DMIF-Network Interface: Is the logical interface at which the DMIF Signalling primitives are identified and mapped into various Signalling messages used on Networks.

DMIF Terminal: Is a terminal where a DMIF Layer is present

DMIF Layer: Is the layer between an Application and the delivery technology that provides the DMIF functions.

DMIF User: Is the applications that exploits the functions offered by the DMIF Layer through the DAI.

Heterogeneous Network: A Network composed of different transport technologies which are connected in tandem through InterWorking Units.

Homogeneous Network: A Network composed of one transport technology only.

Network Session: An association between two DMIF peers providing the capability to group together the resources needed for an instance of a service. The Network Session is identified by a network-wide unique ID. A Network Session could group one or more Service Sessions.

Service: Is an entity identified by a Service Name (opaque to DMIF) which responds to DAI primitives

Service Session: A local association between the local DMIF Layer and a particular service.

A.2 Symbols (and abbreviations)

CAT: Channel Association Tag

DAI: DMIF-Application Interface

DS: DMIF Signalling

DNI: DMIF-Network Interface

QoS: Quality of Service

TAT: Transmux Association Tag

B Authors' Addresses

Vahe Balabanian
Nortel
P.O.Box 3511, St. C
Ottawa, Ontario
Canada K1Y 4H7
Phone: (613) 763-4721
Email: balabani@nortel.ca

C Bibliography

[1] ISO/IEC 14496-1 FCD "MPEG-4 Systems" Oct 1998, obtained from the MPEG Home Page http://drogo.cselt.it/mpeg/

[2] ISO/IEC 14496-2 FCD "MPEG-4 Visual" Oct 1998, obtained from
the MPEG Home Page http://drogo.cselt.it/mpeg/

[3] ISO/IEC 14496-3 FCD "MPEG-4 Audio" Oct 1998, obtained from
the MPEG Home Page http://drogo.cselt.it/mpeg/

[4] ISO/IEC 14496-6 CD "DMIF" Oct 1998, obtained from
the MPEG Home Page http://drogo.cselt.it/mpeg/

[5] Schulzrinne, Lanphier 'Real Time Streaming Protocol
(RTSP)' RFC 2326, Internet Engineering
Task Force, April 1998.

[6] Carsten, Balabanian, Basso, Civanlar, hoffman, Speer,
Schulzrinne, 'RTP payload format for MPEG-4 Elementary
Streams' ietf-avt-rtp-mpeg4-00,
Internet Engineering Task Force, March 1998.

[7] Balabanian, " The Role of DMIF in Support of RTP MPEG-4
Payloads", draft-balabanian-rtp-mpeg4-dmif-00, Sptember 1998.

[8] Balabanian, ' The Use of MPEG-4/DMIF and RSVP with
Integrated Services' draft-balabanian-intserv-mpeg-dmif-00.txt,
Internet Engineering Task Force, Sptember 1998.

Internet Engineering Task Force Audio/Video Transport Working Group
INTERNET DRAFT Balabanian-Nortel
draft-balabanian-rtsp-mpeg4-dmif-00.txt
Sept. 22,1998
Expires: March 26,1999

Original